

Zadanie krajského kola súťaže ZENIT v programovaní

Kategória A a B, 19.11.2019

V prípade nejasností konzultujte záložku **Pomoc** na stránke zenit.ksp.sk, alebo sa spýtajte organizátorov. Úlohy sú hodnotené úplne nezávisle a samostatne, takže ich môžete riešiť v ľubovoľnom poradí. Časový limit označuje, koľko času dostane váš program pri testovaní na našom serveri (nie na vašom lokálnom počítači). Počas súťaže môžete nájsť zadania aj na webstránke. Ak by sa papierové a tlačené zadania v nejakom detaile (napríklad časovom limite) nezhodovali, tak pravdu majú zadania na **webstránke**.

A: Aké číslo si myslím?

10 bodov; časový limit: 300 ms

Ahoj riešiteľ,

Ja som tvoj testovač a myslím si číslo. Tvojou úlohou je napísať program, ktorý vypíše číslo, ktoré si myslím. Program teda nedostane žiadny vstup a má iba jeden pokus vypísať svoj typ. Ak by som ti nepovedal nič viac, bola by to veľmi ťažká úloha. Poviem ti preto, že číslo ktoré si myslím, sa líši od čísla π o dva. Ak tvoj program nevypíše správne číslo, nič si z toho nerob, a skús poslať iný program :).

Tvoj testovač

Vstup a výstup

Na vstupe nič nie je. Na výstup vypíšete svoj typ s presnosťou na dve desatinné miesta. Na oddelenie desatinných miest použijete bodku.

Príklad

vstup

výstup

5 nie je číslo ktoré si testovač myslí. Od čísla π sa líši o menej ako dva.

B: Bujova vláda a Beňova moc

20 bodov; časový limit: 300 ms

V ďalekej krajine za siedmimi predmetmi a viacerými ročníkmi už dlho vládol krutý vládca Buj. Ktokoľvek sa pokúsil poraziť ho a prevziať vládu do vlastných rúk, bol nemilosrdne porazený Bujovými úlohami z algebry, ktoré majú obtiažnosť k .

V dedinke ďaleko od hlavného mesta si žije junák Beňo. Nevediac o tom, je protagonistom napínavého fantas-magorického príbehu (práve ho čítate, anime adaptation TBA) v ktorom sa musí vydať na dlhú výpravu, a napriek všetkým prekážkam a obtiažnostiam nakoniec ukončiť Bujovu krutovládu raz a navždy.

Na to však bude musieť vyriešiť Bujove úlohy z algebry.

Beňova moc je momentálne 1 (preto si myslí, že je obyčajný junák). Jeho skutočná moc je však n^0 . Poctivým čítaním učebníc algebry sa jeho moc zvýši - ako inak, umocní sa, na n^1 . Ak prečíta ďalšiu učebnicu, už bude mať moc n^2 . A tak ďalej.

Aby finále tohto príbehu bolo napínavé, Beňo musí prečítať toľko učebníc, aby sa Bujovi čo najviac vyrovnal, ale neprekonal ho (potom totiž v momente, keď sa zdá že Beňo prehral, mu príde na pomoc starý kamarát Wolfram).

Vstup a výstup

V jedinom riadku vstupu sú dve celé čísla n a k – Beňova moc a obtiažnosť Bujovích úloh z algebry.

Platí $1 \leq n \leq 1\,000$ a $1 \leq k \leq 1\,000\,000$.

Nájdite najväčšie číslo x také, že x je celočíselná mocnina n a $x \leq k$.

Príklad

vstup

výstup

$3^4 = 81$ už nie je ≤ 47 . Beňo teda vie získať najviac $3^3 = 27$ moci.

C: Čavojev kód

25 bodov; časový limit: 300 ms

Čavoj vám poslal zakódovanú správu. Chcete ju odkódovať. Zakódovaný reťazec sa skladá z blokov. Bloky sú binárne reťazce. Každý blok vznikol zakódovaním nejakého čísla. Blokom dĺžky $k \geq 1$ dokážeme zakódovať celé číslo od 0 po $2^k - 1$.

Vstup a výstup

V prvom riadku vstupu je celé číslo k - dĺžka bloku. Všetky bloky sú teda rovnako dlhé. Platí, že $1 \leq k \leq 14$.

V druhom riadku je reťazec s zložený zo znakov 0 a 1 predstavujúci zakódovanú správu. Jeho dĺžka je násobok k a zároveň je najviac 200 000.

Nasleduje 2^k riadkov. V i -tom z týchto riadkov sa nachádza binárny reťazec t_i dĺžky k . To znamená, že číslo i sa zakóduje na t_i . Prvý riadok teda predstavuje kód pre číslo 0, druhý pre číslo 1 a tak ďalej. Môžete predpokladať, že všetky t_i sú navzájom rôzne.

Odkódovaná správa je reťazec znakov, ktorý dostanete tak, že odkódujete s .

Vypíšte odkódovanú správu.

Príklad

vstup

```
3
100011100111100010111001111000
110
011
100
111
101
001
000
010
```

výstup

```
2123273536
```

Prvé 3 znaky reťazca s sú 100. To sa nachádza v riadku reprezentujúcom číslo 2. Prvé číslo výstupu teda bude 2.

D: Dobrá predpoveď

35 bodov; časový limit: 300 ms

Buj sa rozhodol, že si vyskúša predpovedať počasie. Každý z predchádzajúcich n dní spravil m rôznych experimentov alebo pozorovaní, pričom výsledok každého z nich sa dá zosumarizovať ako áno/nie. Ako príklady uvedieme: či vonku svietilo slnko, či bolo oblačno, či bolo vidno mesiac, či boli banány v záhrade zelené, ...

Každý z týchto dní si takisto zaznačil, či pršalo. Teraz by chcel vymyslieť logickú formulu, ktorá mu každý deň bude vedieť predpovedať, či bude pršať alebo nie. (Predstavte si niečo takéto: "Pršať bude práve vtedy, keď je oblačno a zároveň nesvieti slnko.")

Táto formula musí byť samozrejme konzistentná s dátami, ktoré nazbieral. To znamená, že ak ju použije na predpoveď v každom z n pozorovaných dní, dá nám správnu odpoveď.

Nájdite jednu takú formulu, ak existuje.

Vstup a výstup

Na prvom riadku vstupu sú medzerou oddelené dve kladné celé čísla n a m : počet dní a počet experimentov. Platí $n \cdot m \leq 5\,000$. Nasleduje n riadkov, každý popisuje výsledky experimentov v príslušnom dni a tiež či v ten deň pršalo. Formát i -teho riadku je nasledovný:

$$k_{i,1}k_{i,2} \dots k_{i,m} = p_i,$$

pričom $k_{i,j}$ je 0 (výsledok experimentu j je "nie") alebo 1 ("áno"), a p_i je 0 (v daný deň nepršalo) alebo 1 (pršalo).

Ak neexistuje formula konzistentná s týmito pozorovaniami, vypíšte **impossible**. V opačnom prípade na výstup vypíšte ľubovoľnú jednu takú boolovskú formulu. Musí mať presne nasledovnú syntax:

- Boolovská formula je term.
- Term je buď:

- Premenná.
 - Alebo ($t_1 \circ t_2 \circ \dots \circ t_n$) pre $n \geq 1$, kde t_1, t_2 sú termy a \circ je buď `||` (boolovský OR) alebo `&&` (boolovský AND). Ak $n > 2$, tak takýto výraz čítame, že je uzátvorkovaný zľava, t.j. $(a || b \&\& c)$ znamená $((a || b) \&\& c)$.
 - Alebo `!`, kde t je term. Operátor `!` zodpovedá boolovskej negácii.
- Premenná je reťazec začínajúci znakom `x`, za ktorým nasleduje ľubovoľné kladné celé číslo (vzťahujúce sa na výsledok príslušného experimentu). Medzery nemusíte vo formule uvádzať. Vaša formula musí byť dlhá aspoň 1 a nanajvýš 100 000 znakov.

Príklad

vstup	výstup
<pre>5 3 1 0 0 = 0 0 1 1 = 0 0 1 0 = 1 0 0 1 = 1 0 0 0 = 1</pre>	<pre>!((x1) (x2 && x3))</pre>
vstup	výstup
<pre>3 3 1 0 0 = 1 0 1 0 = 1 0 0 1 = 1</pre>	<pre>(x1 x3 !!x2)</pre>
vstup	výstup
<pre>2 1 0 = 0 0 = 1</pre>	<pre>impossible</pre>

E: Ešte raz ... ešte raz

40 bodov; časový limit: 250 ms

Pamätáte si Duplikátnu úlohu zo školského kola? Vašou úlohou bolo napísať program ktorý vypíše program ktorý vyrieši úlohu C. No čo vám poviem, zobrali sme to ešte o krok ďalej.

Vašou úlohou je napísať program, ktorý vypíše program, ktorý vypíše program ktorý vyrieši úlohu C (z krajského kola samozrejme).

Za túto úlohu dostanete toľko bodov, koľko dostane program, ktorý vypíše program ktorý vypíše váš program za úlohu C, ak vypíše fungujúci program.

Vstup a výstup

Na vstupe nič nie je. Na výstup vypíšete program v rovnakom programovacom jazyku, ktorý vypíše program v rovnakom programovacom jazyku, ako je váš odovzdaný program.

Ak vám testovač odpovie WA, môže to znamenať že ešte iba nedokončil všetky 3 fázy testovania a radšej si pozrite detail.

Príklad

vstup	výstup
<pre></pre>	<pre>print('print(5)')</pre>

5 žiaľ nie je správne riešenie žiadneho vstupu úlohy C, teda tento výstup dostane 0 bodov. Výstup je v jazyku Python.

F: Factum fieri infectum non potest

40 bodov; časový limit: 1000 ms

Kubík po dlhoročnom spojenom štúdiu Alchýmie a Informatiky prišiel na zbrusu nový obor - Xorchýmiu. Teda alchýmia za pomoci bitovej operácie xor.

Na obhajobu svojej bakalárky dostal za úlohu predviesť, ako užitočná je táto schopnosť.

Jeho oponent Žaba doniesol na obhajobu n kôpok základných prvkov.

Alchýmia sa zaoberá menením všetkého na zlato. Vie niečo podobné aj táto tvoja pavela, či si si vymyslel, Xorchýmia? Zaútočil Žaba na Kubíkovu bakalárku.

Pomôžte Kubíkovi predviesť moc Xorchýmie!

Úloha

Na stole pred Kubíkom je rozložených n kôpok základných prvkov. V obore Xorchýmie si ich vieme reprezentovať celými číslami od 1 do 5.

V jednom kroku vie Kubík zobrať dve kôpky x a y , a pomocou Xorchýmie dostať jednu kôpku prvku $x \oplus y$. Aby však boli prvky počas celého procesu bezpečné, Kubík nevykoná Xorchýmiu na kôpkach ktoré by vyprodukovali prvok mimo intervalu $[1, 5]$.

Zistite, koľko najviac kôpok ľubovoľného jedného prvku vie Kubík dosiahnuť.

Pripomenka: bitová operácia xor, \oplus , funguje tak že i -ty bit je 1 vo výsledku $x \oplus y$ práve vtedy, keď je 1 v x alebo y , ale nie súčasne.

Vstup a výstup

V prvom riadku je číslo n – počet kôpok. V druhom riadku je n celých čísel p_i – typ prvku v i -tej kôpke.

Platí $1 \leq p_i \leq 5$. V druhej sade $1 \leq p_i \leq 3$.

V prvej sade $1 \leq n \leq 8$. V druhej a tretej $1 \leq n \leq 1000$.

Príklady

vstup

```
4
3 1 2 2
```

výstup

```
3
```

Kubík vie vyxorovať $3 \oplus 1 = 2$, a získať tak 3 kôpky prvku 2.

vstup

```
5
4 5 2 3 3
```

výstup

```
3
```

Tu vie Kubík naxorovať tri trojky.

G: Guľky

45 bodov; časový limit: 1000 ms

Ako ste z názvu iste pochopili, táto úloha je o biliarde. Ste fascinovaný pohybom bilardových guľí a chcete by ste vedieť, kde sa konkrétna guľa nachádza v určitom čase.

V úlohe uvažujeme N guľiek, očíslovaných od 1 po N . Guľky stoja na stole s dĺžkou práve 10^6 metrov (jeho ľavý koniec má súradnicu 0 a pravý 10^6). Na oboch koncoch stola sú bezodné dierky do ktorých guľky vedia spadnúť. Vašou úlohou je vypísať pozíciu jednej špecifikovanej guľky s číslom G po T sekundách.

Keďže toto je ešte jedna z tých ľahších úloh, všetky guľky budú mať rovnakú hmotnosť, všetky budú stáť na jednej priamke, každá guľka bude mať nulový polomer, nebudeme uvažovať rotáciu ani stratu energie a všetky zrážky budú dokonale pružné. Pre dokonale pružnú zrážku dvoch telies, platí nasledovný vzorec:

$$v_1 = \left(\frac{m_1 - m_2}{m_1 + m_2} \right) u_1 + \left(\frac{2m_2}{m_1 + m_2} \right) u_2$$

$$v_2 = \left(\frac{m_2 - m_1}{m_1 + m_2} \right) u_2 + \left(\frac{2m_1}{m_1 + m_2} \right) u_1$$

kde m_i je hmotnosť, u_i je rýchlosť pred zrážkou a v_i je rýchlosť guľe i po zrážke.

Vstup a výstup

Na vstupe máte na prvom riadku číslo N ($1 \leq N \leq 10^5$), počet guľiek. Na druhom riadku je N celých čísel, počiatočné pozície guľiek v metroch. Na treťom riadku je N celých čísel, počiatočné rýchlosti guľiek v metroch za sekundu. (záporná rýchlosť reprezentuje pohyb doľava, kladná rýchlosť pohyb doprava). Štvrtý riadok bude obsahovať dve čísla - G a T . Všetky čísla na vstupe budú v absolútnej hodnote najviac 10^6 .

Na výstup vypíšte jedno číslo, pozíciu guľičky s číslom G v čase T . Výsledok zaokrúhlite na najbližšie celé číslo.

Príklad

vstup

```
1
6
-2
1 4
```

výstup

```
0
```

V čase 3 sa dokotúľa naša guľka na nulový meter, kde spadne do dierky a už tam ostane.

vstup

```
2
1 20
3 -3
2 2
```

výstup

```
14
```

Guľky sa za 2 sekundy nestihnú zraziť a guľka číslo 2 sa za daný čas pohne o 6 metrov doľava.

H: Huncút Hugo

50 bodov; časový limit: 2000 ms

Hugova suseda, teta Hedviga, má na záhrade v rade n trpaslíkov. Hugo, ktorý sa v skutočnosti volá Denis, je huncút a postrach okolia, a každý deň vyvedie Hedvige beťárstvo. Denis sa v noci vkradne na jej záhradu, vyberie si k za sebou idúcich trpaslíkov a usporiada ich podľa výšky od najnižšieho po najvyššieho. Každé ráno to Hedviga zistí, najeduje sa (aj ak sa Hugovim činom poradie nezmení), a napraví ich späť do pôvodného poradia. Denis však nikdy neopakuje svoje gagy (dva gagy považujeme za rovnaké ak poradie trpaslíkov po ich vykonaní je rovnaké). Koľko najviac dní môže Hugo robiť svoje viťúžstva?

Vstup a výstup

Na prvom riadku sú čísla n a k . Platí $1 \leq k \leq n \leq 500\,000$. Na druhom riadku je n čísel oddelených medzerou: nejaká permutácia čísel 1 až n .

Na výstup vypíšete jedno číslo: koľko rôznych permutácií môžeme dostať zo zadanej permutácie tým, že utriedime nejaký súvislý úsek dĺžky k od najmenšieho po najväčší.

Príklady

vstup

```
5 3
1 4 2 5 3
```

výstup

```
2
```

Môžeme dostať $(1, 2, 4, 5, 3)$ a $(1, 4, 2, 3, 5)$.

vstup

```
9 6
9 8 7 6 5 4 3 2 1
```

výstup

```
4
```

Máme štyri možnosti ktorú úsek utriediť, pre každú dostaneme niečo iné.

I: Ignác balil darčeky

55 bodov; časový limit: 1000 ms

Ignác sa rozhodol že tento rok svoje vianočné darčeky zabalí v 2D. Je to vraj jednoduchšie. Aby to mal ešte jednoduchšie, použije iba 2D krabice namiesto 2D papiera, a do nich pozatvára svoje darčeky.

Každý jeden Ignácov darček sa dá popísať niekoľkými bodmi. Pri prechode do 2D sveta sa však body trochu pomiešali a teda na vstupe sú v nejakom poradí a predstavujú tento darček v 2D najlepšie ako sa dá.

Vašou úlohou bude zistiť, akú vysokú 2D krabicu Ignác na darček potrebuje, keď ho otočí tak aby bol čo najnižší. Dôležité je len, aby žiadny bod z krabice netrčal.

Vstup a výstup

Na prvom riadku je celé číslo ($3 \leq n \leq 200\,000$) - počet bodov. Nasleduje n riadkov so súradnicami jednotlivých bodov. Súradnice sú celé čísla medzi -10^9 a 10^9 .

Na výstup vypíšete najmenšiu možnú výšku krabice ktorú treba aby sa do nej objekt zmestil keď ho vhodne otočíme. Číslo vypíšete s presnosťou aspoň na 6 desatinných miest.

Príklad

vstup

```
3
0 0
1 0
0 1
```

výstup

```
0.70710678
```

Na vstupe je pravouhlý trojuholník. Keď ho položíme na preponu, jeho výška je takáto.

vstup

```
4
-1 0
0 0
0 1
-1 1
```

výstup

```
1.000000
```

Na vstupe je štvorec.

J: Jedlo

60 bodov; časový limit: 1000 ms

Keď trávite veľa času hraním počítačových hier, zevraj sa to odrazí na vašom zdraví. Samko sa rozhodol to vykompenzovať tým, že sa bude zdravo stravovať.

Každý vie, že vitamínov je 13: sú to $A, B_1, B_2, B_3, B_5, B_6, B_7, B_9, B_{12}, C, D, E, K$. Vo vyváženej strave musí byť z každého vitamínu aspoň trošku. Samko si teraz ide poskladať svoj denný jedálny lístok. Na výber má z n produktov, každý má nejakú cenu c a obsahuje nejaké látky (nie nutne len vitamíny).

Poradte Samkovi, čo má kúpiť, aby v jeho nákupe boli zastúpené všetky vitamíny a aby ho to vyšlo čo najlacnejšie.

Vstup a výstup

Na prvom riadku vstupe je číslo n : počet produktov. Platí $1 \leq n \leq 3000$.

Na každom z nasledujúcich n riadkoch je popis jedného produktu v tvare $mckl_1l_2\dots l_k$, kde m je názov produktu, c je jeho celočíselná cena v centoch, k je počet látok ktoré obsahuje, a l_i sú názvy jednotlivých látok. Platí $1 \leq c \leq 10^8$, $0 \leq k \leq 20$, všetky názvy pozostávajú z čísel a malých znakov anglickej abecedy a sú dlhé najviac 10 znakov. Názvy produktov sú navzájom rôzne. Vitamíny majú názvy a, b1, b2, b3, b5, b6, b7, b9, b12, c, d, e, k.

Ak neexistuje žiadne riešenie, vypíšte -1 . V opačnom prípade na prvý riadok výstupu vypíšte medzerami oddelenú celkovú cenu produktov a ich počet. Ďalej vypíšte názvy týchto produktov, každý na samostatný riadok. Ak je viacero optimálnych riešení, môžete vypísať ľubovoľné z nich.

Príklad

vstup

```
6
koktejl1 1 3 a b1 b2
koktejl2 1 3 b3 b5 b6
koktejl3 1 3 b7 b9 b12
koktejl4 1 3 c d e
koktejl5 97 1 k
superbomba 100 13 a b1 b2 b3 b5 b6 b7 b9 b12 c d e k
```

výstup

```
100 1
superbomba
```

Draším riešením by bolo kúpiť všetky koktejly, čo by nás vyšlo až 101 centov.

vstup

```
7
koktejl1 1 3 a b1 b2
koktejl2 1 3 b3 b5 b6
koktejl3 1 3 b7 b9 b12
koktejl4 1 3 c d e
koktejl5 97 1 k
bomba1 50 6 a b1 b2 b3 b5 b6
bomba2 50 7 b7 b9 b12 c d e k
```

výstup

```
52 3
bomba2
koktejl2
koktejl1
```

vstup

```
3
hranolky 100 1 sol
hamburger 100 1 sol
kebab 270 1 nacl
```

výstup

```
-1
```

K: Kokos, nech už je po conteste

80 bodov; časový limit: 5000 ms

Máme zadaných n bodov v rovine. Spočítajte koľko existuje rôznych štvorcov, ktoré majú všetky 4 vrcholy v zadaných bodoch a ich strany sú paralelné s osami x a y . Štvorce sú rôzne ak aspoň jeden vrchol majú rôzny.

Vstup a výstup

V prvom riadku je číslo n – počet bodov v rovine. V $(i + 1)$. riadku sú medzerou oddelené 2 čísla x_i a y_i . Sú to súradnice i – teho bodu v rovine. Môžete predpokladať, že všetky zadané body sú rôzne (teda každé 2 body sa líšia aspoň v jednej súradnici).

Na výstup vypíšete jedno číslo a to počet štvorcov s vrcholmi v daných bodoch, ktoré majú všetky strany paralelné so súradnicovými osami.

Vo všetkých vstupoch platí $1 \leq n \leq 100\,000$ a $-10^6 \leq x_i, y_i \leq 10^6$

Príklady

vstup

```
4
0 0
5 5
0 5
5 0
```

výstup

```
1
```

Presne zadané 4 body tvoria štvorec.

vstup

```
4
0 0
1 1
1 -1
2 0
```

výstup

```
0
```

Tu sa štvorec síce nachádza, nemá však strany paralelné s osami.

vstup

```
11
2 2
3 3
2 3
3 2
6 5
6 2
3 5
2 4
4 4
4 2
6 4
```

výstup

```
4
```

